Game AI as Storytelling

Mark Riedl[*]
riedl@cc.gatech.edu

David Thue[†]
dthue@ualberta.ca

Vadim Bulitko[†]
bulitko@ualberta.ca

1 From Rationality To Experience Management

Historically, games have been played between human opponents. However, with the advent of the computer came the notion that one might play with or against a computational surrogate. Dating back to the 1950s with early efforts in computer chess, approaches to game artificial intelligence (AI) have been designed around adversarial, or zero-sum, games. The goal of intelligent game-playing agents in these cases is to maximize their payoff. Simply put, they are designed to win the game. Central to the vast majority of techniques in AI is the notion of optimality, implying that the best performing techniques seek to find the solution to a problem that will result in the highest (or lowest) possible evaluation of some mathematical function. In adversarial games, this function typically evaluates to symmetric values such as +1 when the game is won and -1 when the game is lost. That is, winning or losing the game is an outcome or an end. While there may be a long sequence of actions that actually determine who wins or loses the game, for all intents and purposes, it is a single, terminal event that is evaluated and "maximized." In recent years, similar approaches have been applied to newer game genres: real-time strategy, first person shooters, role-playing games, and other games in which the player is immersed in a virtual world. Despite the relative complexities of these environments compared to chess, the fundamental goals of the AI agents remain the same: to win the game.

There is another perspective on game AI often advocated by developers of modern games: AI is a tool for increasing engagement and enjoyability. With this perspective in mind, game developers often take steps to "dumb down" the AI game playing agents by limiting their computational resources (Liden, 2003) or making suboptimal moves (West, 2008) such as holding back an attack until the player is ready or "rubber banding" to force strategic drawbacks if the AI ever gets the upper hand. The game-playing agent is adversarial but is intentionally designed in an *ad hoc* manner to be non-competitive to make the player feel powerful.

[*]School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA.
[†]Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada.

In this chapter, we focus on game AI which, instead of being designed to win more often, reasons in a *principled* manner about how to make the human player's experience in a game or virtual world more enjoyable. While the outcome of a game is important, it is not the only aspect of a game that a player evaluates. *How* one reaches the ending can often be just as, if not more, important than what the ending is; a hard fought battle that results in a loss can be more enjoyable than an easy win. Extrapolating from the observation that experience can be more important than outcome, we suggest that the goal of computer game AI is to reason about and deliver an enjoyable experience. Game AI thus becomes a tool in the arsenal of the game designer, to be used whenever one would want a real person to play a given role but no one is available. Examples of such roles are:

- Opponents, companions and NPCs that play roles that are not "fun" to play such as shopkeepers, farmers, and victims
- Dungeon master
- Plot writer
- Game designer

As we go down this list, game AI is charged with taking progressively more responsibility for the quality of the human player's experience in the game. To leverage this model, we redefine the task of game AI agents as the creation of an enjoyable player experience, and define payoffs that allow them to optimize the particular qualities of the experience that its designers might desire. Regardless of whether the AI agent is choosing how to oppose or assist the player[1] or how the storyline should unfold, the player's enjoyment is its central concern.

1.1 Reasoning about Experience as Proxy for Designer and Player

We define an experience as one or more interrelated events directly observed or participated in by a player. In games, these events are causally linked series of challenges that play out in a simulated environment (Rollings & Adams, 2003). Intuitively, it is the job of the game designer to make decisions about how to shape the player's experience in a virtual world in order to make it enjoyable. One way game designers do this is by using a "story on rails" to lead players through a dramatically engaging sequence of challenges. While game design approaches have been effective in creating engaging and enjoyable experiences, there is a growing trend toward greater player agency and greater content customization, and neither of these can be achieved easily at the time of game design:

- **Player agency.** Player agency is the ability for the player to do whatever he or she wants at any time. While player agency is typically very high at the action level — the player has the ability to move about the environment and

[1] Roberts et al. (2009) use the term *beyond adversarial* to denote that a game AI system can choose to help or hinder the player based on its assessment of the player's past, current, and future experience.

perform actions — agency at the plot level has typically been restricted to a single storyline or a small number of storyline branches. One reason for this restriction is the combinatorial explosion of authoring storyline branches (Bruckman, 1990; Riedl & Young, 2006); the amount of content that must be authored at least doubles at every branching point, yet a player will only see one branch.

- **Customization.** Players enjoy having opportunities to experience game play that is consistent with their preferred play style (Thue, Bulitko, Spetch, & Wasylishen, 2007). In one study, Thue et al. (2010) demonstrated that the player model-based adaptation of players' in-game experiences resulted in greater reports of fun. The information required to make customization decisions, however, is not available at design time; it must be learned by observing the player during game play.

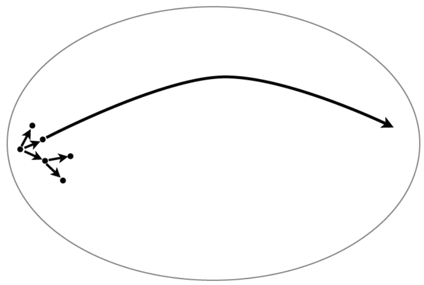
In short, to achieve greater levels of player agency and greater levels of content customization, the computational game system must assume responsibility for the player's experience during play time. The role of determining what the player's experience should be (including how the game world responds to the player's actions) can be delegated to the computational game system itself.

1.2 Leveraging Storytelling

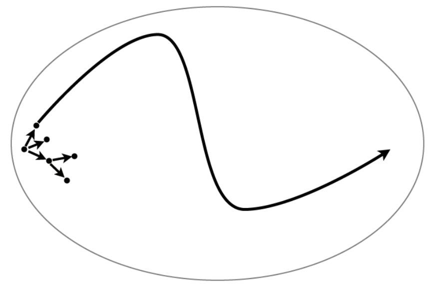
How can an intelligent system in a game or virtual world make decisions about, or indirectly influence, the events that occur in the simulated environment such that the positive qualities of the player's experience are increased? As with game designers, an intelligent system can leverage correlations between experience and narrative to reason about how to manage a player's experience. A narrative is the recounting of a sequence of events with a continuant subject and that constitutes a whole (Prince, 1987). Thus, both "experience" and "narrative" are descriptions of sequences of events. From a game design perspective, an experience is a description — at some level of abstraction or specificity — of events that are *expected* to unfold.

An *Experience Manager* — a generalization of the concept of *Drama Manager* first proposed by Laurel (1986) and first investigated by Bates and colleagues (cf., Bates, 1992; Kelso, Weyhrauch, & Bates, 1993) — is an intelligent system that attempts to coerce the state of the world such that a structured narrative unfolds over time without reducing the perceived agency of the interactive player. An Experience Manager uses the principle of narrative to *look ahead* into possible futures of the player's experience to determine what *should* happen in the world over time to bring about an enjoyable, structured experience. The projection of a narrative sequence into the future enables the Experience Manager to evaluate the global structure of possible player experiences in a way that cannot be achieved by looking at any single world state in isolation. The question that must be addressed is: in light of an interactive player, how can a computational system project a narrative into the future toward maximizing the positive qualities of the player's experience?

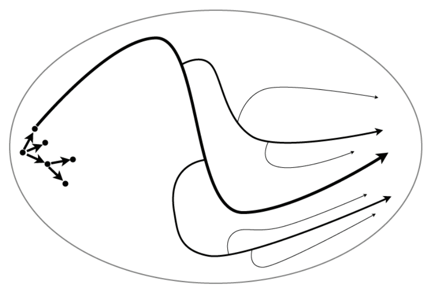
In artificial intelligence, problems are often modeled as state spaces, where every point in this abstract space is a particular configuration of the game environment.



(a) A possible trajectory through state space.



(b) A trajectory that visits areas of state space, yielding one with certain properties.



(c) Accounting for player interaction.

Figure 1: The Experience Management problem is to compute trajectories through state space.

Whereas a game playing agent may attempt to maximize expected payoff by choosing which state to transition to next, a system attempting to optimize player experience must choose a *sequence* of states through which the game should transition. We refer to a sequence of state transitions as a *trajectory* through state space. Choosing a trajectory is non-trivial; even when the state space is finite, the number of possible trajectories can be infinite when loops are allowed, as is often the case with stories. Figures 1(a) and 1(b) show two possible trajectories through state space. Every point in the oval is a possible state configuration for the entire virtual world. Actions, performed by the player or the computational system (possibly through the actions of non-player characters), cause transitions from one world state configuration to the next; in any given state, there are potentially many actions that can be performed. If we assume that states of the right side of the oval are terminal states, then the trajectory in Figure 1(a) may be one that reaches a terminal state in the fewest transitions, a metric classically used to determine the efficiency and optimality of a solution. The trajectory in Figure 1(b), however, may enter parts of state space that, when taken together, are more interesting, dramatic, or pedagogically meaningful. The challenge is to computationally find the one trajectory in the space of all trajectories that will optimize the player's experience.

Because a narrative is the recounting of a non-random sequence of events, *any trajectory through a state space is a narrative*. Making the connection between a trajectory of states in a game and a narrative enables us to cast the search for a particular trajectory as the problem of generating a story. The computational generation of stories is still an open problem; existing story generation algorithms exist that are capable of identifying *expected* sequences of events in a game or virtual world, although not at a level comparable to human creative performance. Further, the problem of story generation is made more complex by the fact that an interactive player has the ability to act in the world, making it impossible to guarantee that any narrative sequence will unfold as expected. That is, if the player, knowingly or inadvertently, performs actions that cause the virtual world state to deviate from the expected trajectory, the next best trajectory must be computed, as show in Figure 1(c). In that sense, managing the player's experience is the problem of searching for many alternative stories. In this article, we describe a technique for Experience Management that employs story generation technologies to manage an interactive player in a virtual world. The goal of the approach is to coerce the events in a virtual world such that the player has an experience with certain well-defined narrative properties.

The heart of Experience Management is the tension between meaningful player agency and the desire to bring about a narrative experience that is coherent and conforms to the designer's pragmatic and aesthetic ideals (Riedl, Saretto, & Young, 2003). Having considered the Experience Manager as a proxy for the designer, we may also consider the extent to which the player's preferences are part of the definition of the "optimal" narrative trajectories. A player model informs the trajectory search about what the player will find interesting and enjoyable, such that when there are multiple ways of achieving the designer's pragmatic and aesthetic requirements, the trajectory that appeals most to the player can be selected.

In this chapter, we discuss how game AI can be reinterpreted as storytelling for the purpose of reasoning about the human player's experience, thereby creating greater player agency through more meaningful interactions, and affording more customization of experience. We begin with an overview of *narrative intelligence* — the ability to reason about narrative — and narratological foundations of computational storytelling. In Section 3 we provide a computational representation of narrative and narrative construction. In Section 4 we describe how an Experience Manager can use the ability to computationally generate narrative to manage the player's interactive experience in a game or virtual world. Finally, in Section 5, we address customization of the player's interactive experience through learning a model of the player and using it to optimize his or her particular narrative trajectory.

2 Narrative Intelligence and Narratological Foundations

Narrative and storytelling are terms that are widely understood but not often well defined. The definition of narrative used in this chapter is: *the recounting of a sequence of events that have a continuant subject and constitute a whole* (Prince, 1987). Narrative as entertainment, in the form of oral, written, or visual storytelling, plays a central role in many forms of entertainment media, including novels, movies, television, and theatre. Narrative is also used in education and training contexts to motivate and to illustrate.

One of the reasons for the prevalence of storytelling in human culture may be due to the way in which narrative is a cognitive tool for situated understanding (Bruner, 1990; McKoon & Ratcliff, 1992; Gerrig, 1993, 1994; Graesser, Singer, & Trabasso, 1994). There is evidence that suggests that we, as humans, build cognitive structures that represent the real events in our lives using models similar to the ones used for narrative in order to better understand the world around us (Bruner, 1990). Our understanding of the world is achieved by "constructing reality" as a sequence of related events from our senses (Bruner, 1991). Whereas we tend to understand inanimate objects through cause and effect, we attempt to understand the intentional behavior of others through a sophisticated process of interpretation with narrative at its core (Bruner, 1990). This *narrative intelligence* (Blair & Meyer, 1997; Mateas & Sengers, 1999) is central in the cognitive processes that we employ across a range of experiences, from entertainment contexts to active learning.

Narratologists break narrative down into at least two layers of interpretation: *fabula* and *sjuzet* (Bal, 1998). The *fabula* of a narrative is an enumeration of all the events that occur in the story world between the time the story begins and the time the story ends. The events in the fabula are temporally sequenced in the order that they occur, which is not necessarily the same order in which they are told. The *sjuzet* of a narrative is a subset of the *fabula* that is presented via narration to the audience. If the narrative is written or spoken word, the narration is in natural language. If the narrative is a cinematic presentation, computer game, or virtual world, the narration is through the actions of characters and the camera shots that capture that action. While it is the narrated *sjuzet* that is directly exposed to the audience, it is the *fabula* of a narrative that is the content of the narrative (*i.e.*, what the narrative is about).

In this article we focus on *fabula*: what happens (or what is expected to happen) in the virtual world or game. Readers interested in how stories can be computationally structured at the *sjuzet* level should see Montfort (2007), Cheong and Young (2008), Bae and Young (2008), and Jhala (2009).

There are many aspects that determine whether a story is accepted by the audience as "good." Many of these aspects are subjective in nature, such as the degree to which the audience empathizes with the protagonist. Other aspects appear to be more universal across a wide variety of genres. Cognitive psychologists have determined that the ability of an audience to comprehend a narrative is strongly correlated with the causal structure of the story (Trabasso & Sperry, 1985; van den Broek, 1988; Graesser, Lang, & Roberts, 1991; Graesser et al., 1994) and the attribution of intentions to the characters that are participants in the events (Graesser et al., 1991; Gerrig, 1993; Graesser et al., 1994). Story comprehension requires the perception of causal connectedness of story events and the ability to infer the intentions of characters. Accordingly, we assert that two nearly universal qualities of narratives are *logical causal progression* and *character believability*.

The causality of events is an inherent property of narratives and ensures a whole and continuant subject (Chatman, 1993). Causality refers to the notion that there is a relationship between temporally ordered events such that one event changes the story world in a particular way that enables future events to occur (Trabasso & van den Broek, 1985). For a story to be considered successful, it must contain a degree of causal coherence that allows the audience to follow the logical succession of events and predict possible outcomes. One can think of the property of logical causal progression as the enforcement of the "physics" of the story world in the sense that there are certain things that can and cannot happen based on the actual state of the story world and the characters within it. For example, in fairy tales, the world is such that wild animals such as wolves can eat people without killing them.

Character believability (Bates, 1994) is the audience perception that arises when the actions performed by characters do not negatively impact the audience's suspension of disbelief. Goal-oriented behavior is a primary requirement for believability (Loyall, 1997; Charles et al., 2003). Specifically, we, as humans, ascribe intentionality to agents with minds (Dennett, 1989). The implication is that if a character is to be perceived as believable, one should be able to, through observations of the character, infer and predict its motivations and intentions. For a greater analysis of goal-directed behavior in character believability see Riedl and Young (2010).

3 Computation of Narrative Structure

Addressing Experience Management as story generation, there are two problems to consider. The first is how to computationally model narrative structure. The second is how to computationally model the process of constructing narrative and managing interactive experiences. This section addresses the computational representation of narrative in detail, but only touches on algorithms for generation due to the fact that there are still many open research problems that remain to be addressed.

The general consensus among psychologists and computer scientists is that a narrative can be modeled as a semantic network of concepts (Trabasso, Secco, & van den Broek, 1984; Graesser et al., 1991; Young, 1999; Swartjes & Theune, 2006). Nearly all cognitive representations of narrative rely on causal connections between story events as one of the primary elements that predict human narrative comprehension. Following others (Lebowitz, 1987; Young, 1999; Riedl et al., 2003; Riedl & Young, 2004; Porteous & Cavazza, 2009; Riedl & Young, 2010), we employ AI plan-like representations of narrative as transitions through state space. Plan representations have been used in numerous narrative intelligence systems because they correlate well with the narratological and cognitive constructs that have been associated with narrative reasoning.

3.1 Narrative as Plans

Partial-order causal link (POCL) plans (Weld, 1994), in particular, have been used successfully to computationally reason about narrative structure because of strong correlations between representation and cognitive and narratological concepts (Young, 1999; Christian & Young, 2004). A POCL plan is a directed acyclic graph in which nodes are operations (also called actions) which, when executed, change the world state. Arcs capture causal and temporal relations between actions. A *causal link*, denoted $a_i \rightarrow^c a_j$, captures the fact that the execution of action a_i will cause condition c to be in true in the world and that that condition is necessary for the execution of subsequent action a_j. Causal links, unique to POCL plans, are representationally significant due to the importance of causality in narratives. *Temporal links* capture ordering constraints between

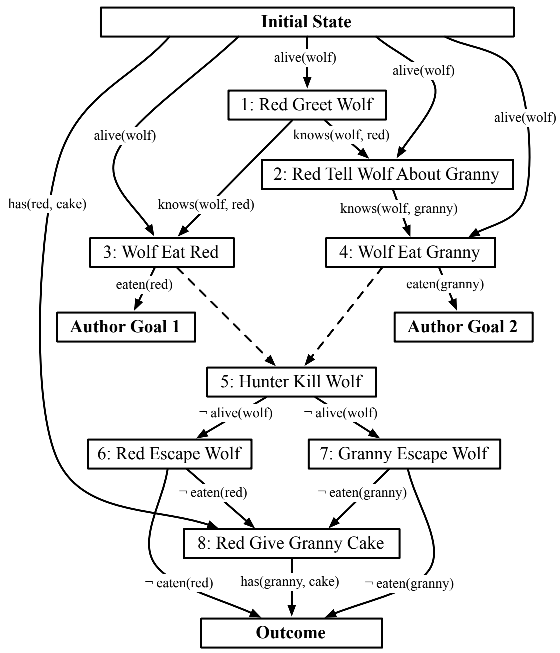


Figure 2: The story of Little Red Riding Hood represented as a partially ordered plan.

actions when one action must be performed before another. Temporal and causal links create a partial ordering between actions, meaning that it is possible that some actions can occur during overlapping time intervals.

A narrative is a sequence of events — significant changes to the state of the story world. The mapping of narrative to plan is straightforward. Events are represented by plan actions, which are partially ordered with respect to each other by the temporal links. The term "event" captures the nuance that not all changes to the world state are intentional on behalf of some agent or character. Thus, some events can be accidents, automatic reactions to other changes, and forces of nature. Partial ordering is a favorable feature of a story representation because it is often the case that actions in the *fabula* occur simultaneously. In the remainder of this article, we will use the terms "action" and "event" interchangeably. Note that a narrative plan for an interactive game or virtual world contains events to be initiated by the player and non-player characters.

Figure 2 shows an example of a plan representing a narrative sequence for a simplified version of *Little Red Riding Hood* (Grimm & Grimm, 1812). Boxes represent events. Solid arrows are causal links where the labels on the links describe the relevant conditions. Dashed arrows represent temporal constraints between events. For clarity not all causal and temporal links are shown. There are three types of special constructs shown in the figure. The *initial state* is a description of the story world as a set of logical propositions. The initial state specifies characters in the story world, the properties of characters, and relationships between characters, props, and the world. The *outcome*

Action: Eat (?wolf, ?victim)

```
Precondition:  wolf(?wolf), person(?victim), alive(?wolf), alive(?victim),
               ¬eaten(?wolf), ¬eaten(?victim)
Effect:  eaten(?victim), in(?victim, ?wolf), full(?wolf)
```

Action: Tell-About (?speaker, ?hearer, ?topic)

```
Precondition:  character(?speaker), character(?hearer), alive(?speaker),
               alive(?hearer), ¬eaten(?speaker), ¬eaten(?hearer),
               knows(?speaker, ?hearer), knows(?speaker, ?topic),
               ?speaker ≠ ?hearer
Effect:  knows(?hearer, ?topic)
```

Figure 3: Portion of a domain library for Little Red Riding Hood.

is a description of how the story world should be different after the story completes. In Figure 2, the outcome is that the Granny character has cake, Granny is not in the state of being eaten and Little Red Riding Hood ("Red" for short) is also not in the state of being eaten. Finally, *author goals* are intermediate states that must be achieved as some point during the course of the story. In the example, the two author goals are that Granny becomes eaten by something and Red becomes eaten by something. We use author goals to preserve the authorial intent of the designer, as described in Section 3.2.

Next, we overview how plans are computationally constructed. Planners are search algorithms that solve the planning problem: given an initial world state, a domain theory, and a goal situation, find a sound sequence of operators that transforms the world from the initial state into a state in which the goal situation holds. The *domain theory* describes the "physics" of the world — how the world works and how it can be changed. The domain theory is often a library of action (or event) templates where applicability criteria and world state update rules are specified through logical statements. Specifically, events have *preconditions* and *effects*. The precondition of an event is a logical statement that must be true in the world for the event to legally occur. The effect of an event is a logical statement that describes how the world would be different if the event were to occur. Figure 3 shows two event templates from the domain library for the Little Red Riding Hood world.

There are many algorithms that solve the planning problem. In this section we highlight *partial-order planning* (POP) (Penberthy & Weld, 1992; Weld, 1994). POP planners are refinement search algorithms, meaning they inspect a plan, identify a flaw — a reason why the current plan being inspected cannot be an actual solution — and attempt to revise the plan to eliminate the flaw. The process iteratively repairs one flaw at a time until no flaws remain. It is often the case that there is more than one way to repair a flaw, in which case the planner picks the most promising repair, but remembers the other possibilities. Should it make a mistake, the planner can *backtrack* to revisit any previous decision point. We favor POP because the particular way in which flaws are identified and revised in POP is analogous to cognitive planning behavior in adult humans when faced with unfamiliar situations (Rattermann et al., 2001).

The refinement search process starts with an empty plan. A flaw is detected, and zero or more new plans are produced in which the flaw is repaired (and often introducing new flaws). These plans become part of the fringe of a space consisting of all possible complete and incomplete plans. The process is repeated by picking the most promising plan on the fringe and iterating.

In POP, there are two types of flaws: *open conditions*, and *causal threats*. An open condition flaw exists when an event (or the outcome) has a precondition that has not been satisfied by the effect of a preceding event (or the initial state). An open condition flaw is repaired by applying one of the following strategies:

1. Select an existing event in the plan that has an effect that unifies with the precondition in question.

2. Select and instantiate an event template from the domain library that has an effect that unifies with the precondition in question.

A causal threat flaw occurs when the temporal constraints do not preclude the possibility that an action a_k with effect $\neg c$ can occur between a_i and a_j when there is a causal link $a_i \rightarrow^c a_j$ requiring that c remain true. Causal threats are repaired by adding additional temporal constraints that force a_k to occur before a_i or after a_j. By iteratively repairing flaws, the current plan progressively gets closer to a solution. The algorithm terminates when it finds a plan that has no flaws. More details on POP are provided by Weld (1994).

3.2 Preserving Designer Intent

A narrative generator assumes responsibility for the structure of the player's experience during gameplay. It is, however, desirable that a designer is able to constrain the space of possible experiences the player can have to enforce a particular aesthetic or pragmatic vision. We extend the standard POCL plan representation to include *author goals* (Riedl, 2009), partially-specified intermediate states that the story must pass through at some point before the story concludes. Potential solutions that do not satisfy each author goal state description at some point between the initial state and the end state are pruned from consideration. Author goals serve two important purposes. First, author goals constrain the narrative search space such that it is impossible for a generator to produce a story that does not meet certain criteria imposed by the designer[2]. Second, author goals can be used to force complexity in the story.

The importance of author goals as part of the narrative representation becomes clear in the context of the Little Red Riding Hood example. Without author goals, achieving the outcome — Granny has cake, Granny is not eaten, and Red is not eaten — is trivial. Red need only give some cake to Granny, which can be achieved with a single event. The author goals — Granny is eaten, and Red is eaten — force the story generator to figure out how to have both Granny and Red eaten and then saved.

[2] In the absence of a well-defined evaluation function that can rate the "goodness" of a narrative trajectory, the designer's intent is the only guidance the story generator has.

3.3 Generating Believable Stories

If plans are good representations of narratives, might it also make sense to use planning algorithms to construct narratives? Young and Saver (2001) provide neurological evidence of functional similarity between planning and narrative generation in the human brain. Planning algorithms, however, are general problem solvers that make strong assumptions about the nature of the problem being solved. Specifically, a planner is an algorithm that attempts to find a sequence of operations that transforms the world state from an initial configuration into one in which a given goal situation holds. While a resultant set of operations – a plan – can be considered a narrative, that narrative is unlikely to be believable or to contain aesthetic features such as a dramatic arc that would be favorable for the task of creating engaging experiences (Riedl & Young, 2010). The reason that conventional planners are not guaranteed to generate believable narrative plans is because of their emphasis on achieving valid plans; they disregard the requirement that characters will appear motivated by intentions other than the author's goals. Even with a heuristic that favors plans in which characters appear believable, it is possible for a conventional planner to return a plan that is not believable when it finds a shorter, valid solution before it finds a longer, valid, and believable solution.

To reliably generate narrative plans in which characters appear believable, narrative planners must utilize new definitions for plan completeness that include believability, coupled with mechanisms for selecting actions that move the planner toward complete, believable solutions. Extensions to POP, implemented in the FABULIST story generation system (Riedl & Young, 2010) allows planners to search for narrative sequences that are both logically and causally coherent but also present events that explain the underlying motivations of characters. This is one step toward computationally achieving the "illusion of life" necessary for suspension of disbelief (Bates, 1994). Future work in story generation must also consider aesthetics such as dramatic arc — the cadence of rising action, climax, and falling action — and suspense. While efforts are underway to explore computational reasoning about such story aesthetics (cf., Fitzgerald, Kahlon, & Riedl, 2009; Appling & Riedl, 2009), there are many open research questions to be addressed in the pursuit of computational systems that can assume full responsibility for the quality of a player's interactive experience. In the remainder of this chapter, we will describe our approach to Experience Management in the context of simple POP, although more sophisticated algorithms exist that are keyed to the specific problem of generating believable narrative sequences.

4 Experience Management

Player experience in a virtual world or game can be expressed as a narrative, projecting an ideal trajectory of state transitions into the future. This narrative is *not* necessarily the sequence of moves that a rational computer opponent would take to maximize expected payoff but rather the one that delivers a "good" experience to the player. In a virtual world modeled after Little Red Riding Hood, this may be the sequence that raises the stakes for the player but then allows the player to overcome adversity to save the day. In a game of chess, this may be the sequence that sets up a dramatic come-from-behind victory. Thus far, however, we have not addressed the fact that the player is not just another character in the story, but a human with his or her own goals and the ability to make gameplay choices that differ from the idealized narrative sequence. That is, the human player neither knows the script, nor is expected to follow it.

Experience management is the process whereby a player's agency is balanced against the desire to bring about a coherent, structured narrative experience. On one hand, we want the player to have the perception that he or she has the ability to make decisions that impact the world in a meaningful way (e.g., at the plot level). On the other hand, the designer wants the player to have an experience that meets certain aesthetic and pragmatic guidelines. Can we allow meaningful player agency while still achieving the goal of bringing about an experience that has the features desired by the designer? Designers of heavily plot-driven computer games often resort to a "story on rails" approach, where although there may be an *appearance* of agency, the world is structured so that a single pre-scripted plot sequence unfolds; side-quests are then often added to enhance the player's feeling of agency. The "story on rails" approach is diametrically opposite of simulation style games, in which there is no pre-scripted plot sequence and any narrative structure emerges from the interactions of autonomous non-player characters and human players.

Our approach to experience management, as implemented in the AUTOMATED STORY DIRECTOR framework (Riedl et al., 2008) and the MIMESIS system (Young et al., 2004), balances player agency and narrative structure by allowing meaningful player agency and then *generating* novel narrative trajectories when the player, intentionally or inadvertently, exerts their agency. Consider trajectory space — the set of all possible trajectories through state space. One trajectory, the *exemplar trajectory*, is the human designer's preferred story; it is the best possible experience according to that designer. The exemplar trajectory projects the player's actions and non-player character actions into the future. Players may still exert their agency, however, and we categorize their actions as follows (Riedl et al., 2003):

- **Constituent** — the player knowingly or unknowingly performs the action that is listed as the next action in the narrative. For example, after the Wolf has eaten Red and Granny, the player, in the role of the hunter, kills the Wolf.
- **Consistent** — the player performs an action that is not part of the narrative but does not significantly alter the state of the world and the narrative sequence can continue. For example, early in the game, the player talks to Red.
- **Exceptional** — the player performs an action that is not part of the narrative and the world state is changed such that some portion of the narrative cannot continue. For example, the player kills the Wolf before the Wolf meets Red, or the player takes the cake away from Red.

In the case that the player performs an exceptional action, the Experience Manager must figure out how to allow the player's action[3] and still achieve an experience with the requisite structure. Note that the exceptional player action may not immediately

[3] One can also consider attempting to prevent the exceptional action in a natural and unobtrusive manner. Riedl et al. (2003) describe a technique called *intervention* whereby the exceptional action is surreptitiously

threaten the narrative, as the change to the world may impact an action that is projected to occur far downstream in time. Handling an exceptional player action is tantamount to finding the next best trajectory given a world state altered by that action.

4.1 Anticipating Necessary Narrative Plan Adaptations

Using plan structures to model narrative is advantageous because, by capturing the causal relationships between actions, a narrative plan can be analyzed for points in which exceptional player actions are possible. That is, assuming the narrative plan executes as expected, we can look into the future and identify possible exceptional actions. We use a technique similar to that described by Riedl et al. (2003) to analyze the causal structure of the scenario to determine all possible inconsistencies between plan and virtual world state that can occur during the entire duration of the narrative. Inconsistencies arise due to exceptional player actions performed in the world. The technique identifies intervals of the narrative plan during which it is possible for an exceptional action to occur.

For every possible inconsistency that can arise that threatens a causal link in the plan, an alternative narrative plan is generated. For each possible inconsistency that can arise, we use the following repair process to find an alternative trajectory. First, we assume the narrative will progress as expected until the threatened interval begins. Next, we assume the worst case: that the player will perform the exceptional action that creates the inconsistency. By simulating the execution of the exceptional player action, we can infer the state that the world would be in if the action were to occur. Finally, the following repair processes are tried in order until one succeeds in generating a narrative that meets the designer's intent:

(i) The threatened causal link is removed, leaving an open condition flaw on the terminus event, and the planner is invoked.

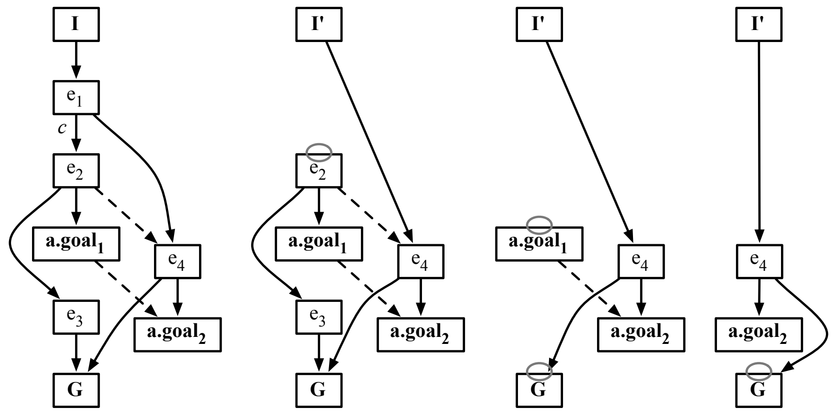
(ii) The threatened causal link is removed, the terminus event and all other events (except author goals) that are causally downstream (e.g. there is a path from the threatened causal link to a given event through the directed graph of causal links) are removed, open conditions flaws are identified on the remaining events, and the planner is invoked.

(iii) The threatened causal link is removed, the terminus event and all other events (including author goals) that are causally downstream are removed, open conditions flaws are identified on the remaining steps, and the planner is invoked.

(iv) The remaining plan is discarded, a new outcome situation and new author goals are selected, and the planner is invoked.

---

replaced by a nearly identical action that does not affect the world state in a way that prevents the narrative from progressing. For example, if shooting a character prevents that character from performing a critical task in the future, then *shoot* can be replace by *gun-jam* that prevents the character from dying and allowing the narrative to continue. However, if intervention is chosen, it effectively removes player agency, which may or may not be noticed by the player.



(a) Exemplar (b) Tier i (c) Tier ii (d) Tier iii

Figure 4: Illustration of the tiered re-planning strategies considering a single possible inconsistency resulting in $\neg c$ in exemplar (a). Figures (b) – (d) show how the exemplar is prepared for re-planning for each tier.

To illustrate the tiers of repair strategies, consider the narrative plan in Figure 4(a). Event e_1 establishes condition c in the world, which is necessary for event e_2. Suppose it is possible for the player to perform an action that causes $\neg c$ to become true during the interval between the completion of e_1 and the beginning of e_2. The possible inconsistency is found during causal analysis, and the tier (i) strategy is invoked. A copy of the plan is made and updated to reflect the state the world would be in should all events preceding the interval in question have occurred. That is, the initial state now represents the world state after e_1 has occurred, and action e_1 is no longer part of the narrative. The tier (i) strategy removes only the causal links in the interval in question that are threatened by the exceptional player action. Figure 4(b) shows the copy of the plan after tier (i) pre-processing but before the story generator is invoked to fill back in causally necessary events. Ovals indicate flaws in the plan due to pre-processing. Re-planning will most likely result in the insertion of new events prior to e_2 that reestablish c in the world.

Suppose that the tier (i) strategy fails; the story generator cannot find any partially ordered sequence of new events that can fill the gap created by removing the threatened causal link. The Experience Manager advances to the tier (ii) strategy, and removes threatened causal links, the events satisfied by the threatened links, and all events that are causally downstream except author goals. A causally downstream event is any action e_i such that there is a path in the graph of causal links from a removed event to e_i. In this example, action e_3 is causally downstream but e_4 is not. Figure 4(c) shows the copy of the plan after tier (ii) pre-processing. The tier (iii) strategy is similar to tier (ii) except that causally downstream author goals are also removed. The underlying assumption is that tier (ii) failed because the author goals were interfering with the ability of the story generator to find a valid plan. Figure 4(d) shows the copy of the plan after tier (iii) pre-processing. Finally, should all other strategies fail, the tier (iv) strategy (not shown) deletes all actions in the plan, replaces the outcome situation G with a new outcome situation G', and instantiates any number of new author goals. The new outcome and author goals come from a list of alternative author goals specified at design time by the human designer. If the final tier of replanning fails, we resort to a non-managed virtual world, relying completely on game play dynamics and the autonomy of non-player characters to create an emergent narrative experience.

We use the tiered strategy approach to compensate for the fact that story planners are not yet at human-level ability for story creation. In the absence of a story planner that can reliably evaluate the "goodness" of a narrative sequence, the tiered strategy approach is built on the assumption that the human-authored exemplar narrative is the ideal experience and that any necessary changes should preserve the original narrative structure as much as possible. This assumption is not true in all cases, and can result in situations where the Experience Manager attempts to undo the consequences of the player's actions (Riedl et al., 2008). As story generation techniques improve (see Section 3.3 for pointers to potential improvements), reliance on such assumptions will become unnecessary, simplifying the operation of the Experience Manager.

4.2 Computation of Contingencies

Story replanning is performed offline to avoid delays due to computation (Riedl et al., 2003, 2008); for any sufficiently rich world, the online generation of narrative structure can exceed acceptable response times in an interactive game or virtual world. The result of the offline replanning process is a tree of contingency plans in which each plan represents a complete narrative starting at either the initial world state (for the exemplar) or at the point in which an inconsistency can occur at play time. If the player performs an exceptional action, the system simply looks up the appropriate branch in the tree of contingencies and seamlessly begins using the new trajectory to manage the player's experience from that point on. The contingency tree is necessary for dynamic execution; by pre-generating the tree, an Experience Manager can rapidly switch to alternative narrative plans when player actions make this necessary. Riedl and Young (2006) show that the contingency tree is functionally equivalent, but more expressive, than a choose-your-own-adventure style branching story. The additional expressivity comes from the fact that player actions can be performed at any time (e.g., in any interval). Note that a tree of contingency plans can be potentially infinite in depth. We use a simple user model to determine which exceptional actions are most probable, and focus on making those parts of the tree more complete. Additionally, as a matter of practicality, we cap the depth to which the tree can grow.

Figure 5 shows a portion of the contingency tree automatically generated for the Little Red Riding Hood domain. As before, inside the plan nodes solid arrows represent causal links and dashed arrows represent temporal constraints. The vertical i-beams alongside each plan node represent intervals during which exceptional actions can occur and result in inconsistencies that need to be handled. The arrows between plan nodes indicate which contingency narrative plan should be used if an inconsistency does in fact occur during interactive execution. The actual contingency tree for even the simple Little Red Riding Hood world requires thousands of contingencies, most of which are minor variations of each other (see Section 4.3 for execution details).

For online narrative plan execution, events in the current narrative plan are interpreted as abstract descriptions at the level commonly associated with plot. The events are used to generate directives to an underlying execution system that regulates game play. Each event can be thought of as a subset of the virtual world's overall state space. The execution system may or may not include semi-autonomous characters (Mateas & Stern, 2000; Riedl et al., 2008). We point the interested reader to details on the *Automated Story Director* framework (Riedl et al., 2008) for specifics on one possible execution system.

4.3 Example: Little Red Riding Hood

Our approach to experience management, as implemented in the AUTOMATED STORY DIRECTOR framework (Riedl et al., 2008), is illustrated in an interactive experience based loosely on the *Little Red Riding Hood* tale. The virtual world was built on a MOO (a text-based, object-oriented, multi-user dimension). Figure 6 shows a screenshot of the Little Red Riding Hood story in execution.

The player assumes the role of the Hunter (in the screenshot in Figure 6, the player has chosen the name Fred). Although the Hunter is not the title character, the hunter is the character that ultimately "saves the day." Note that experience management can be performed regardless of which character the player controls. Experience management works best in rich virtual worlds with many characters. To make the Little Red Riding Hood domain more suitable for experience management, we extended the domain to include two extra characters: a fairy, and a monster named Grendel. The fairy has the power to resurrect dead characters. Grendel, like the Wolf, is capable of swallowing other characters alive.

In this simple example, the principal way in which the player expresses his or her agency is through the act of killing other characters at times other than that specified in the current narrative trajectory. A portion of the contingency narrative plan tree is shown in Figure 5, only showing a few interesting branches (for space, plans are truncated to show only the actions that occur before the author goals). The exemplar narrative plan is the root of the contingency tree, shown at the left of the figure. Consider the narrative plan node labeled 1. To reach this trajectory, the player must create an inconsistency by killing the Wolf before it can eat either Red or Granny. The simplest alternative trajectory is to have the Fairy resurrect the Wolf, who then continues as normal. If for some reason the Fairy is also killed by the player, Grendel can fill the role of the character who eats Red and Granny, achieving the author goals. Note that in the exemplar narrative, the plot points specifying that the Wolf eat Red and that the Wolf eat Granny are unordered with respect to each other. This creates the possibility of multiple branches based on a race condition between the player's killing of the Wolf and the achievement of the two author goals: the Wolf can be killed before eating Red or Granny (contingency plan 1); the Wolf can be killed after eating Red but before eating Granny (contingency plan 2); or the Wolf can be killed after eating Granny but before eating Red (contingency plan 3). Each possible ordering of events in the race

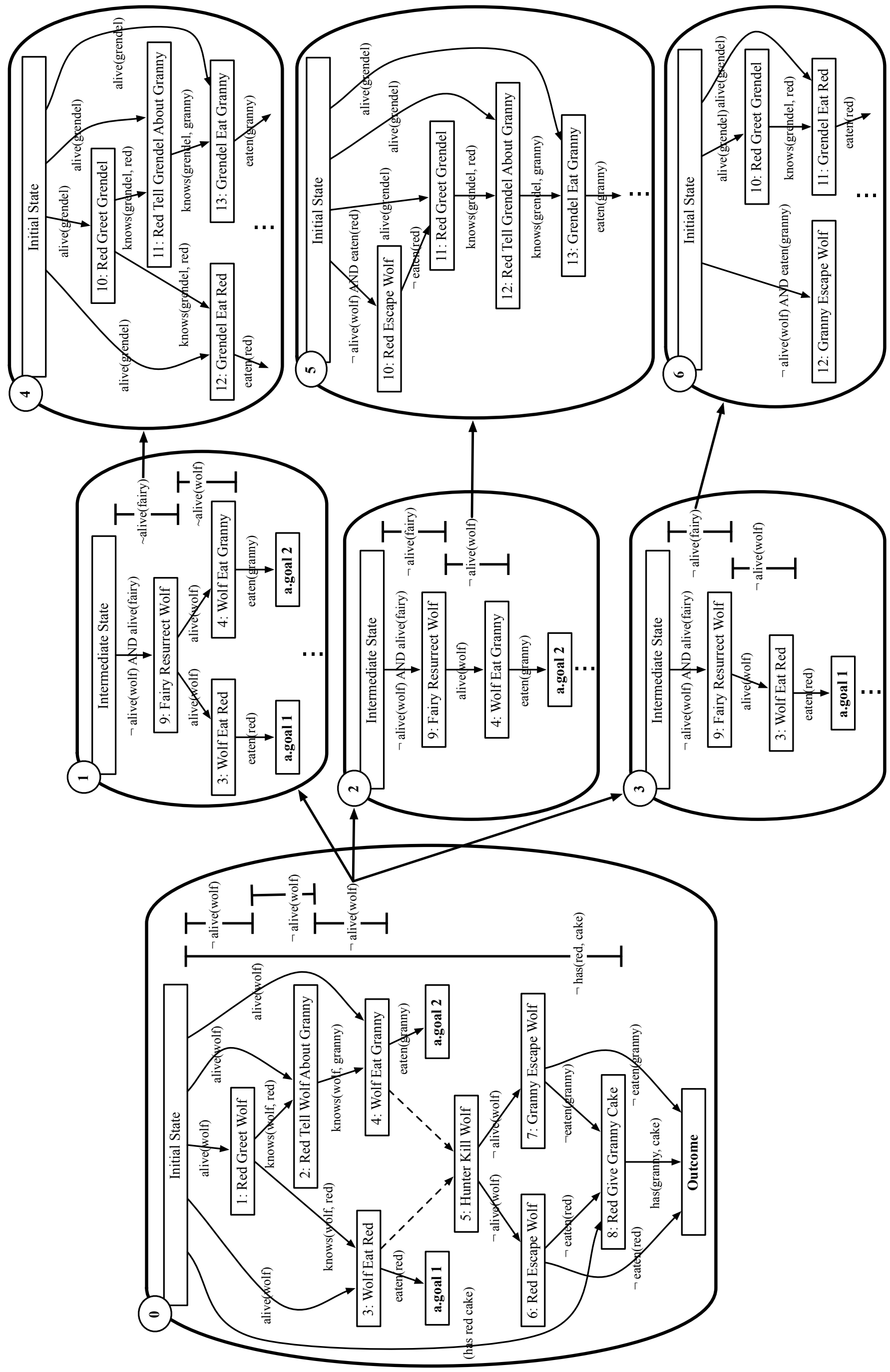


Figure 5: Part of the tree of narrative plan contingencies for Little Red Riding Hood.

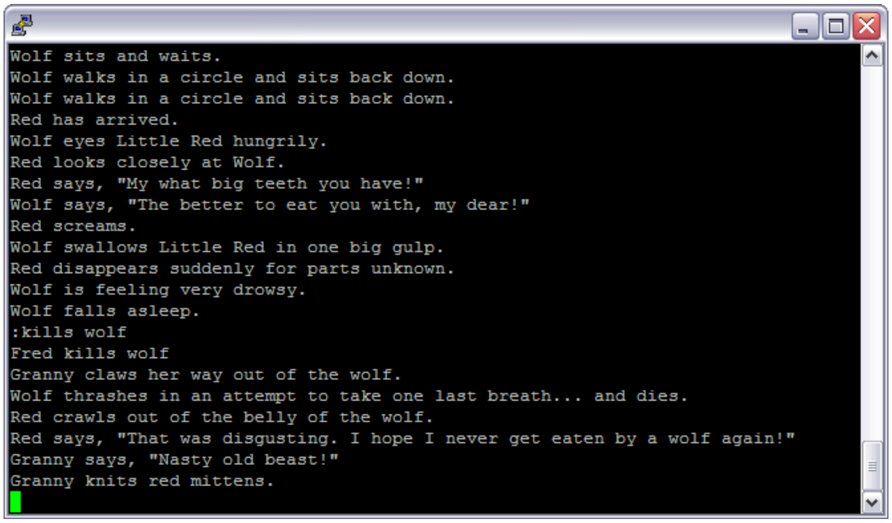


Figure 6: Screenshot of the Little Red Riding Hood interactive story in a MOO.

condition results in a slightly different narrative trajectory.

The Little Red Riding Hood domain and exemplar narrative shown in Figure 2 results in 1319 branches when the contingency tree is generated to a depth of 5 (the root of the tree, the exemplar narrative, is at depth 0). With a depth of 5, the contingency tree can handle five exceptional player actions in one play session before reverting to an emergent, unmanaged world. It takes approximately 43 minutes (approximately 11 minutes spent on garbage collection) to generate the contingency tree on an Intel Core2 Duo 3GHz system with 3GB of RAM and 100GB of virtual memory running ALLEGRO CL® 8.0.

5 Player Modeling

Having considered the Experience Manager as a proxy for the designer, we may also consider the extent to which the player's preferences are part of the definition of "optimal" narrative trajectories. Building and using models of player behavior is becoming increasingly prevalent in commercial video games, as doing so enables a computational game system to learn about the player in order to make decisions that impact the player's experience in a positive way. Player modeling in games has been used to maximize coherence, interest, and enjoyment.

- **Maximizing coherence of player experience.** A player model can be used to predict when the player might perform actions that diverge from the expected sequence and respond appropriately (Magerko, 2006; Harris & Young, 2010).
- **Maximizing interest.** Learning player preferences over plot points and other narratively salient situations allows a game system to present an experience that

is customized to the player's interests (Barber & Kudenko, 2007; Sharma, Mehta, Ontanón, & Ram, 2007; Li & Riedl, 2010).

- **Maximizing enjoyment.** Learning player preferences over style of play has been shown to translate directly toward more engaging, enjoyable experiences (Seif El-Nasr, 2007; Thue et al., 2010).

The greater the Experience Manager's knowledge of its audience, the more informed its decisions about the player's experience will be. Due to the dynamic nature of games and virtual worlds, the point at which the Experience Manager has the *most* information about the player is just before a decision needs to be made. These facts motivate both learning and using a player model regularly during the course of the player's experience, and we present these tasks as two computational challenges for an Experience Manager to overcome: learning a profile of the player, and effectively utilizing this model to positively affect the player's experience.

5.1 Learning About the Player

One promising approach, as implemented in the PASSAGE system (Thue et al., 2007, 2010), is to learn about the player regularly throughout their interactive experience. One advantage of this approach, as opposed to learning about the player *before* game play begins, is that if the player's preferences change as the experience unfolds, the player model can be refined. Specifically, we propose to learn the player's preferences toward different styles of play (Thue et al., 2007), drawn from Laws' (2001) theory for providing entertaining pen-and-paper role playing games. Laws identifies the following play styles:

- Fighter (f) — for players who enjoy engaging in combat;
- Method Actor (m) — for players who enjoy having their personality tested;
- Storyteller (s) — for players who enjoy considering complex plots;
- Tactician (t) — for players who enjoy thinking creatively; and
- Power Gamer (p) — for players who enjoy gaining special items and abilities.

Thus, a player model is a vector of scalars, $\langle f, m, s, t, p \rangle$, describing the extent to which the player has exhibited the traits of each play style. To determine whether the player is exhibiting a particular play style, player actions in the domain theory are annotated as being indicative of different styles of play; whenever the player performs an action that has been annotated, the corresponding value in the model increases. The player model is thus an estimate of the player's inclinations toward playing in each of the modeled styles.

5.2 Using a Player Model

Given the goal of maximizing player enjoyment, we can leverage the primary assumption of the PASSAGE system (Thue et al., 2007): that players will enjoy events which allow them to play in their modeled play-styles more than events which favor other styles of play. Annotations on events (those performed by players and NPCs) indicating the play style that they are most suited for link the player model to the real time execution of a narrative sequence. Thus the player model, represented as a vector of play style preference strengths, acts as a metric for each sequence to calculate its expected utility. This calculation could be as simple as examining the distribution of actions in the narrative sequence based on their annotations as to which play styles they support. For example, with a model of $\langle f = 1, m = 0, s = 0, t = 1, p = 2\rangle$, then the ideal narrative for this player would be made up of a collection of actions, distributed such that 25% appeal to fighters, 25% to tacticians, and 50% to tacticians. In the event that a narrative is not ideal for a player, the expected utility will be some value in the interval $[0, 1]$ indicating appropriateness based on event annotations.

Previously, we considered *how* to determine whether a player action is exceptional or not. We now consider *why* the player performs an exceptional action. There are many reasons why exceptions occur, including ignorance of the plotline, accident, malicious behavior (ie., trying to "break" the game)[4], or expression of a style of play that differs from the expected play style. If the exception occurs because of an expression of a particular play style, we wish to optimize the player's experience by repairing the narrative, accommodating the player's action into the narrative structure, and making any adjustments necessary to increase the expected utility of the subsequent narrative plan. Because the Experience Manager cannot know the precise configuration of the player model until the moment the exception is executed, narrative branches that account for many different possible configurations of the player model must be generated prior to the game being played. Fortunately, the generation of narrative branches is one of the key features of the approach to Experience Management described earlier.

We extend our Experience Management approach to utilize the real time dynamics of the player model in the following way. Instead of sequentially working through the four tiers of re-planning strategies (see Figure 4) escalating only when one of the strategies fails, the system executes all four strategy tiers for every possible inconsistency. Further, we modify tier (iv) to draw from many different sets of alternative author goals instead of selecting the next best set. We propose this because author goals force the story planner to consider trajectories that pass through different portions of state space (Riedl, 2009). In the absence of a human-level story generator, using sets of varied author goals forces the Experience Manager to explore a wider variety of trajectories, whereas without guidance, the planner may err on the side of making the fewest changes that it can.

The modifications described above result in one or more alternative branches for any given possible inconsistency. For example, if there are three sets of alternative author goals, then the maximum number of alternative branches per possible inconsis-

[4] The "cooperative contract" of interactive entertainment (Young, 2002) suggests that if a player is not interested in being entertained in the way the game was designed, the designer need not be responsible for entertaining the malicious player.

tency is seven: one from tier (i), one from tier (ii), one from tier (iii), and three from tier (iv). See Fig. 7 for an illustration of branching execution incorporating a player model. The figure introduces decision nodes (diamonds) that select a child node (a narrative plan) based on the player model. In the illustration, killing the Wolf before event 1 has completed results in a possible inconsistency — the Wolf is unable to complete the action or any subsequent actions — and the possible inconsistency can be repaired in one of three ways.

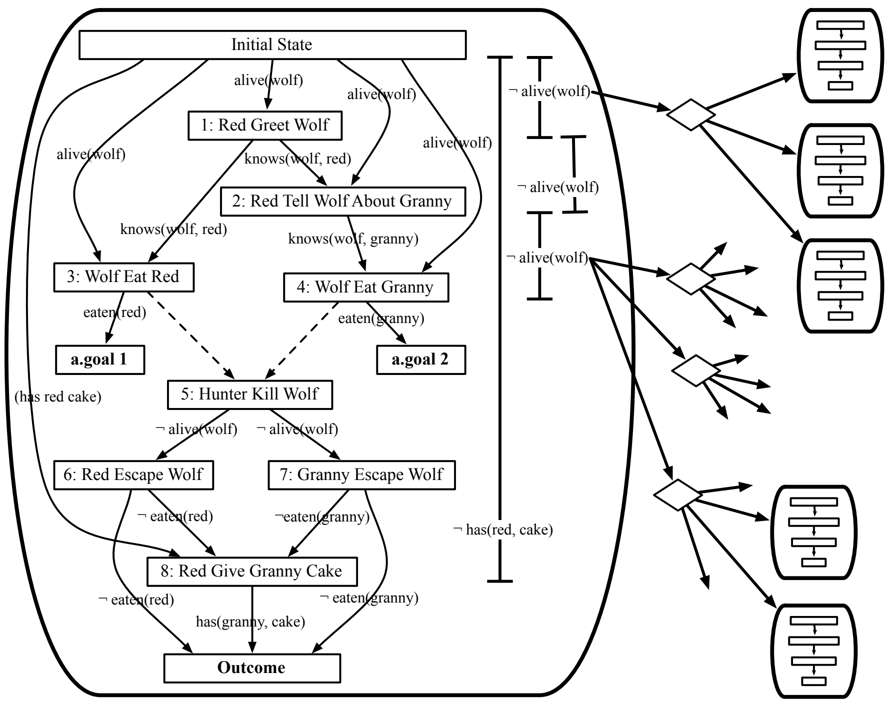


Figure 7: A portion of a tree of narrative plan contingencies with decision nodes for inspecting the real time player model.

When an inconsistency arises due to an exceptional player action, the system knows definitively that the current plan cannot continue to execute; the only non-determinism at this point is which branch to take. The system calculates the expected utility of each branch based on the configuration of the player model at the time the exceptional player action occurs. Thus the branch that actually begins execution at the time of the exceptional user action both restores causal coherence and also tunes the player's experience according to his or her preferred play style.

The Experience Management approach incorporating real time dynamics of the player model increases the number of narrative contingencies that must be generated *a priori*. However, the increase is by a linear number of branches per branching point and thus does not significantly increase the computational complexity of building the tree of contingency narratives. There is also an additional burden placed on the human designer in the sense that he or she must now provide as many sets of author goals as possible. Event templates for the world domain must also be annotated according to style of play. These additional authorial requirements are deemed relatively negligible and future advancements in story generation will be less reliant on authorial guidance from humans. The added benefit is that there are multiple contingencies available for every possible inconsistency, meaning that the Experience Manager can optimize the narrative trajectory the player is on, with respect to the choices available. The extensions to Experience Management, merging AUTOMATED STORY DIRECTOR and PASSAGE, have not been implemented; however, we believe the combination of story generation-based experience management and player model to be a promising means of addressing player agency and customization.

6 Conclusions

Experience management is the process whereby a player's agency is balanced against the desire to bring about a coherent, structured narrative experience. Intuitively, this is what game designers do when they construct a game world with a narrowly prescribed set of paths that deliver the player to a satisfying conclusion. However, due to the growing trend toward greater player agency and greater content customization, we must consider computational approaches that offload design and management of the player's game play experience onto automated computational systems. In this work, we present an approach to automated, real time experience management, in which we leverage the correlations between narrative and experience. By generating narrative trajectories that project possible experience into the future, a system is able to coerce a game or virtual world so that designer intent is preserved without diminishing player agency. The system is also able to reason about the narrative trajectory that maximizes the player's enjoyment based on acquired information about the preferences of the player toward certain styles of play. Thus, when the player exerts his or her agency in ways that are inconsistent with the provided narrative structure, the system is capable of seamlessly recovering *and* bringing the narrative trajectory in line with the player's inferred desires.

This perspective on how Artificial Intelligence can be used to create engaging gameplay expands the traditional role of an AI agent from adversarial opponent — focused on maximizing payoff over time (e.g., beating the player) — to an agent with the goal of increasing the player's enjoyment. While there are many open research questions that remain with regard to generating better stories, the experience management framework suggests that whenever the global experience of a computer game is more important than achieving any one terminal state, be it a non-narrative game like chess or a game highly driven by plot, modeling the AI as storytelling is a beneficial approach.

References

Appling, D. S., & Riedl, M. O. (2009). The role of plot understanding in plot generation. In Louchart, S., Roberts, D., & Mehta, M. (Eds.), *Intelligent Narrative Technologies II: Papers from the 2009 Spring Symposium (Technical Report SS-09-06)*, Palo Alto, CA. AAAI Press.

Bae, B.-C., & Young, R. M. (2008). A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. In *Proceedings of the 1st International Conference on Interactive Digital Storytelling*.

Bal, M. (1998). *Narratology: An Introduction to the Theory of Narrative*. University of Toronto Press.

Barber, H., & Kudenko, D. (2007). Dynamic generation of dilemma-based interactive narratives. In *Proceedings of the 3rd Artificial Intelligence and Interactive Digital Entertainment conference*.

Bates, J. (1992). Virtual reality, art, and entertainment. *Presence: The Journal of Teleoperators and Virtual Environments*, *1*(1).

Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, *37*(7), 122–125.

Blair, D., & Meyer, T. (1997). Tools for an interactive virtual cinema. In Trappl, R., & Petta, P. (Eds.), *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents*. Springer.

Bruckman, A. (1990). The combinatorics of storytelling: Mystery train interactive. Available: http://www.cc.gatech.edu/ asb/papers/combinatorics-bruckman-90.pdf (retrieved 1 Nov. 2007).

Bruner, J. (1991). The narrative construction of reality. *Critical Inquiry*, *18*(1), 1–21.

Bruner, J. (1990). *Acts of Meaning*. Harvard University Press, Cambridge.

Charles, F., Lozano, M., Mead, S., Bisquerra, A., & Cavazza, M. (2003). Planning formalisms and authoring in interactive storytelling. In *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*.

Chatman, S. (1993). *Reading Narrative Fiction*. Macmillan Publishing Company, New York.

Cheong, Y.-G., & Young, R. M. (2008). Narrative generation for suspense: Modeling and evaluation. In *Proceedings of the 1st International Conference on Interactive Digital Storytelling*.

Christian, D., & Young, R. (2004). Comparing cognitive and computational models of narrative structure. In *Proceedings of the 19th National Conference on Artificial Intelligence*.

Dennett, D. (1989). *The Intentional Stance*. MIT Press, Cambridge, MA.

Fitzgerald, A., Kahlon, G., & Riedl, M. O. (2009). A computational model of emotional response to stories. In *Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling*.

Gerrig, R. (1994). Narrative thought?. *Personality and Social Psychology Bulletin*, *20*(6), 712–715.

Gerrig, R. J. (1993). *Experiencing Narrative Worlds: On the Psychological Activities of Reading*. Yale University Press, New Haven.

Graesser, A., Lang, K. L., & Roberts, R. M. (1991). Question answering in the context of stories. *Journal of Experimental Psychology: General*, *120*(3), 254–277.

Graesser, A., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review*, *101*(3).

Grimm, J., & Grimm, W. (1812). Little red cap. In *Kinder- und Hausmärchen, 1st ed*, Vol. 1. Berlin.

Harris, J., & Young, R. M. (2010). Proactive mediation in plan-based narrative environments. *IEEE Transactions on Computational Intelligence and AI in Games*, *in press*.

Jhala, A. H. (2009). *Cinematic Discourse Generation*. Ph.D. thesis, North Carolina State University.

Kelso, M., Weyhrauch, P., & Bates, J. (1993). Dramatic presence. *Presence: The Journal of Teleoperators and Virtual Environments*, *2*(1).

Laurel, B. (1986). *Toward the Design of a Computer-Based Interactive Fantasy System*. Ph.D. thesis, Ohio State University.

Laws, R. (2001). Robin's laws of good GMing. Steve Jackson Games.

Lebowitz, M. (1987). Planning stories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*.

Li, B., & Riedl, M. O. (2010). An offline planning approach to game plotline adaptation. In *Proceedings of the 6th Conference on Artificial Intelligence for Interactive Digital Entertainment Conference*.

Liden, L. (2003). Artificial stupidity: The art of intentional mistakes. In *AI Game Programming Wisdom*, Vol. 2. Charles River Media.

Loyall, A. B. (1997). *Believable Agents: Building Interactive Personalities*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

Magerko, B. (2006). *Player Modeling in the Interactive Drama Architecture*. Ph.D. thesis, Computer Science and Engineering, University of Michigan.

Mateas, M., & Stern, A. (2000). Towards integrating plot and character for interactive drama. In Dautenhahn, K. (Ed.), *Social Intelligent Agents: The Human in the Loop: Papers from the AAAI Fall Symposium (Technical Report FS-00-04)*. AAAI Press, Menlo Park, CA.

Mateas, M., & Sengers, P. (1999). Narrative intelligence. In Mateas, M., & Sengers, P. (Eds.), *Narrative Intelligence: Papers from the 1999 Fall Symposium (Technical Report FS-99-01)*. AAAI Press, Menlo Park, CA.

Seif El-Nasr, M. (2007). Interaction, narrative, and drama creating an adaptive interactive narrative using performance arts theories. *Interaction Studies*, *8*(2).

McKoon, G., & Ratcliff, R. (1992). Inference during reading. *Psychological Review*, *99*.

Montfort, N. (2007). Ordering events in interactive fiction narratives. In Magerko, B., & Riedl, M. (Eds.), *Intelligent Narrative Technologies: Papers from the AAAI Fall Symposium (Technical Report FS-07-05)*. AAAI Press.

Penberthy, J. S., & Weld, D. S. (1992). UCPOP: A sound, complete, partial-order planner for ADL. In *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*.

Porteous, J., & Cavazza, M. (2009). Controlling narrative generation with planning trajectories: the role of constraints. In *Proceedings of the 2nd International Conference on Interactive Digital Storytelling*.

Prince, G. (1987). *A Dictionary of Narratology*. University of Nebraska Press, Lincoln.

Rattermann, M., Spector, L., Grafman, J., Levin, H., & Harward, H. (2001). Partial and total-order planning: evidence from normal and prefrontally damaged populations. *Cognitive Science*, *25*, 941–975.

Riedl, M. O. (2009). Incorporating authorial intent into generative narrative systems. In Louchart, S., Roberts, D., & Mehta, M. (Eds.), *Intelligent Narrative Technologies II: Papers from the 2009 Spring Symposium (Technical Report SS-09-06)*, Palo Alto, CA. AAAI Press.

Riedl, M. O., Saretto, C., & Young, R. M. (2003). Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multi-Agent Systems*.

Riedl, M. O., Stern, A., Dini, D. M., & Alderman, J. M. (2008). Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on System Science and Applications, Special Issue on Agent Based Systems for Human Learning and Entertainment*, *3*(1).

Riedl, M. O., & Young, R. M. (2004). An intent-driven planner for multi-agent story generation. In *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems*.

Riedl, M. O., & Young, R. M. (2006). From linear story generation to branching story graphs. *IEEE Journal of Computer Graphics and Animation*, *26*(3), 23–31.

Riedl, M. O., & Young, R. M. (2010). Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, *in press*.

Roberts, D. L., Riedl, M. O., & Isbell, C. (2009). Beyond adversarial: The case for game AI as storytelling. In *Proceedings of the 2009 Conference of the Digital Games Research Association*.

Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on Game Design*. New Riders.

Sharma, M., Mehta, M., Ontanón, S., & Ram, A. (2007). Player modeling evaluation for interactive fiction. In *Proceedings of the AIIDE 2007 Workshop on Optimizing Player Satisfaction*.

Swartjes, I., & Theune, M. (2006). A fabula model for emergent narrative. In *Proceedings of the 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment*.

Thue, D., Bulitko, V., Spetch, M., & Wasylishen, E. (2007). Interactive storytelling: A player modelling approach. In *Proceedings of the 3rd Artificial Intelligence and Interactive Digital Entertainment Conference*.

Thue, D., Bulitko, V., Spetch, M., & Webb, M. (2010). Socially consistent characters in player-specific stories. In *Proceedings of the Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Trabasso, T., & Sperry, L. (1985). Causal relatedness and importance of story events. *Journal of Memory and Language*, *24*, 595–611.

Trabasso, T., Secco, T., & van den Broek, P. (1984). Causal cohesion and story coherence. In Mandl, H., Stein, N., & Trabasso, T. (Eds.), *Learning and Comprehension in Text*. Lawrence Erlbaum Associates.

Trabasso, T., & van den Broek, P. (1985). Causal thinking and the representation of narrative events. *Journal of Memory and Language*, *24*, 612–630.

van den Broek, P. (1988). The effects of causal relations and hierarchical position on the importance of story statements. *Journal of Memory and Language*, *27*, 1–22.

Weld, D. (1994). An introduction to least commitment planning. *AI Magazine*, *15*.

West, M. (2008). Intelligent mistakes: How to incorporate stupidity into your AI code. *Game Developer Magazine - Digital Edition*.

Young, K., & Saver, J. L. (2001). The neurology of narrative. *SubStance: A Review of Theory and Literary Criticism*, *30*, 72–84.

Young, R. M. (1999). Notes on the use of plan structures in the creation of interactive plot. In Mateas, M., & Sengers, P. (Eds.), *Narrative Intelligence: Papers from the AAAI Fall Symposium (Technical Report FS-99-01)*. AAAI Press, Menlo Park.

Young, R. M. (2002). The co-operative contract in interactive entertainment. In Dautenhahm, K., Bond, A., Canamero, L., & Edmonds, B. (Eds.), *Socially Intelligent Agents*. Kluwer Academic Press.

Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R., & Saretto, C. (2004). An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, *1*, 51–70.